

区间dp

2019年4月9日 18:56

- ◆
- ◆ 区间dp

一. 区间dp

1. 阶段一般与区间长度有关，维度可以由左右端点描述

- ◆
- ◆ 例题

二. 剪彩

1. 题意：1~n个点的权值分别是v，剪去v[i]时，幸运值增加v[l]*v[i]*v[r]，其中l是左边和右边第一个没被剪掉的点，左或右结点被剪光时视作1，求剪得出的最大幸运值
2. 变量d[l][r]存储(l,r)剪光而l和r留着时的最大幸运值
 - i. 最左和最右有两个剪不动的结点，可以用v[0]=v[n+1]=1解决，则d[0][n+1]就是答案
3. 阶段len：区间长度，此处为了方便循环设len为r-l，len从2到n+1
4. 状态l，r：固定len时l和r的值，此处为了方便循环设l=r-len，r从len到n+1
5. 决策k：(l,r)之间的每个可以剪的结点（且设为最后一个被剪的结点）
6. 转移：d[l][r]=l<k<r时的max{ d[l][k]+d[k][r]+v[l]*v[k]*v[r] }
 - i. 此处的求max既可以让d初值0，循环的求max(d,转移)
 - ii. 也可以新开一个t，求完max(t,转移)后让d=t
7. v[0]=v[n+1]=1;
for__(len,2,n+1) //r-l的长度
for__(r,len,n+1){
int l=r-len;
for__(k,l+1,r)
d[l][r]=max(d[l][r], d[l][k]+ d[k][r]+ v[l]*v[k]*v[r]);
cout<<d[0][n+1];

三. 不太典型的区间dp：求n个数分成m个区间，区间和的乘积的最大值（USSTD5B）

1. 变量d[i][j]存储花了i天，分完j个数的最大值
2. 阶段i：天数，从i-1转来即可，似乎i应在l前
3. 状态l：这一天从l+1开始选
4. 状态r：这一天帮完r人
5. inline int sm(int l,int r){ //s[r]-s[l] = sum of (l,r)
return s[r]-s[l];}
6. d[0][0]=1;
for__(i,1,m) //i天结束后
for__(l,0,n) //左端点：前一天帮完l人
for__(r,l+1,n) //右端点：今天帮完r人
d[i][r]= max(d[i][r], d[i-1][l] *sm(l,r));
7. 所有转移顺序：（l的起点i-1全都可以改成0）
 - i. for__(i,1,m) for__(l,i-1,n) for__(r,l+1,n)

- ii. for__(i,1,m) for__(r,1,n) for_(l,i-1,r)
- iii. for_(l,0,n) for__(r,l+1,n) for__(i,1,m)
- iv. for__(r,1,n) for_(l,0,r) for__(i,1,m)
- v. for_(l,0,n) for__(i,1,m) for__(r,l+1,n)
- vi. for__(r,1,n) for__(i,1,m) for_(l,i-1,r)